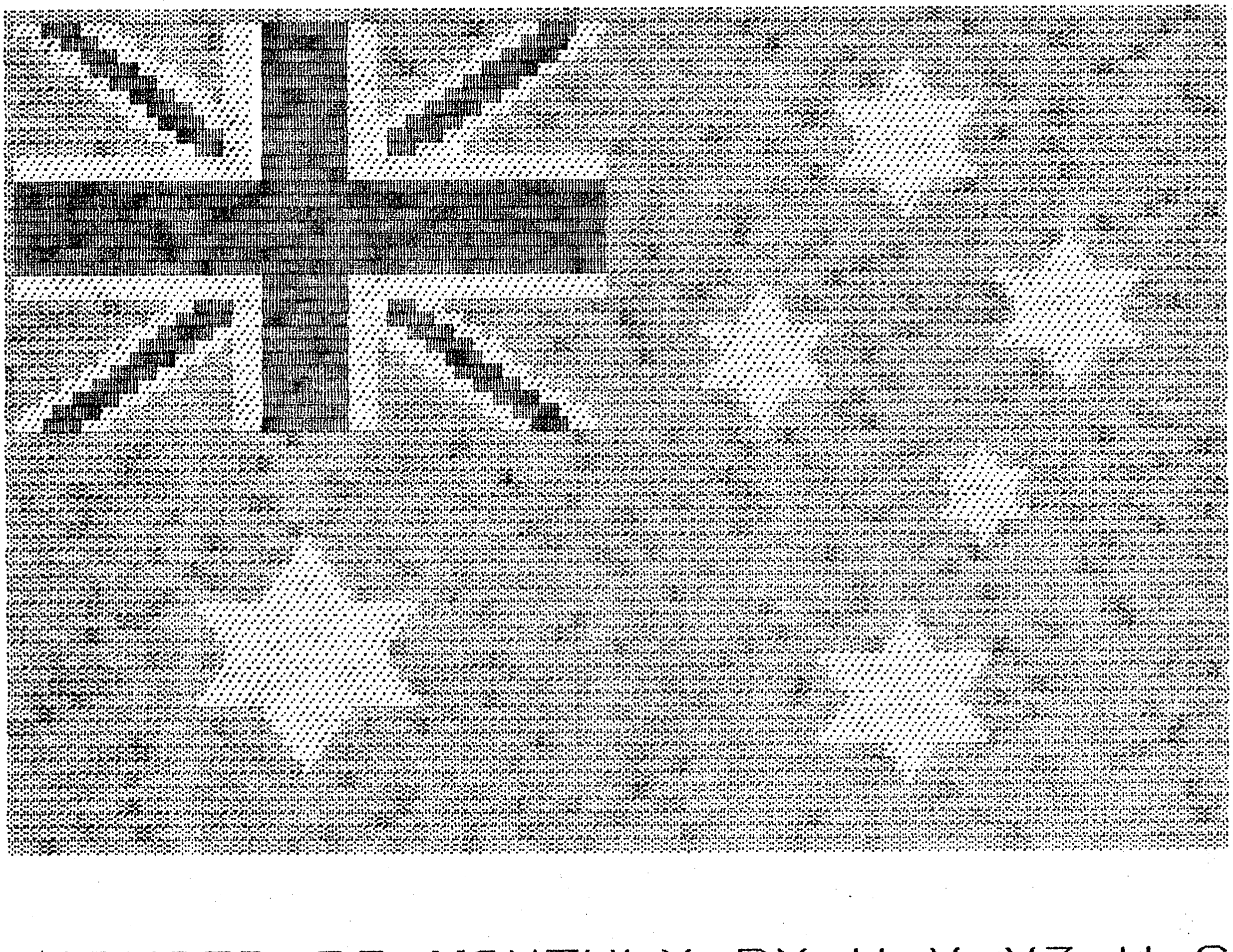
SEPT/OCT 1988 ISSUE # 20

VZ 200/300

HUNTER VALLEY VZ JOURNAL

CELEBRATING THE BICENTENARY



PRODUCED BI-MONTHLY BY H.V.VZ.U.G.

EDITOR'S COMMENTS ISSUE # 20 - 2

ERRORS IN PUT/GET * 3 IN ISSUE 19, Page 4 :-
Lines 2 to 8 must not be typed in as it will corrupt program. I entered lines to provide a heading without realising program would be affected. My apologies to Robert Quinn andour readers.

See SUITE in this issue for an enlarged and enhanced PUT/GET * 3.

APOLOGIES, EXPO 88 & BRISBANE AND OTHER VZ USER GROUPS :- Page 3

PAGED DISK MENU by Paul & Joe LEON :- Pages 4-5
This disk menu is very similiar to one published previously. It will handle up to 120 programs which will be displayed on up to five pages.

SUITE by Robert QUINN :- Pages 6-7
This is an enlarged and enhanced PUT/GET * 3 chief of them being IBS (INTERRUPT BSAVE SCREEN) for disk users.

PRINTER BUFFER UPDATE by Dave BOYCE :- Page 8
Dave continues updating with new information on the buffer.

32K RAM / 64K EPROM from David NEWCOMBE :- Pages 9-10
With the VZ being discontinued a source of hardware or software is essential to VZ Users. David's 32K Ram Board and or 64K Eprom module might be just what you need. Please check for prices and availability first.

60 * 60 DIGIT MULTIPLIER by Neville HUGHES :- Page 11
If you're into maths then this program might prove interesting.

BEAM HEADING by Des HOLMES :- Pages 12-13
This program is for radio hams. Although it's centered on Newcastle it should be possible to change it for your location.

4K-64K SIDEWAYS RAM/EPROM by Joe LEON :- Pages 14-15
In the VZ memory map there is 10K reserved for cartridges from 4000H to 67FFH. The DOS ROM in the disk controller takes up 8K which leaves us with a 2K gap which can be utilised. The 2K can't be enlarged normally exept by bank switching (going sideways), up to 128K possible.

MUSICAL JOYSTICKS by Brendon GREGG :- Page 15
A sipmle program by Brendon which will allow you to play a tune on your VZ. The sound commands can be adjusted as required. Take it away MAESTRO.

FOR PRIVATE SALE :- Page 15
Dave has some more goodies for sale surplus to his requirements.

FOR SALE E & F W.P. PATCH 3.1 :- Page 15
NEW - NEW - PATCH 3.3 - NEW - NEW - All I can tell you at this stage is it has extra commands/functions like UPDATE, ENHANCED RENAME AND ERASE, SOFTWARE SHIFTLOCK AND PRINTER CONTROL CODES IMBEDDED IN TEXT.

SOFTWARE FOR SALE :- Page 16
Versatile utilites by Dave mitchell and Matthew Taylor.

HUNTER VALLEY VZ JOURNAL SUPPLEMENT :- 5 Back Pages

RAM COMMUNICATION REGION UPDATE by Robert QUINN - Original by Dave MITCHELL

Another first for the Journal, a pull out supplement with original idea by Brian Greeve. The idea is to remove the supplement pages from previous and this issue and staple them together to form the start of a reference library. A front cover has been provided for that purpose with provision for an index.

APOLOGIES :-

Every now and then we can't help our readers problems simply because we are not familiar with the subject or we just don't have the particular program in question they have trouble with in using. We try to enlist other VZ users who may be able to help and that is as far as we can go.

BRISBANE VZ USERS WORKSHOP & EXPO 88 -

EXPO 88 was real great and so was the local VZ User Group who made me most welcome in their club and homes as well. It was a real pleasure meeting in person Larry Taylor, John D'Alton, Bob Jones and Eddie Tomes among others.

VZ DOWN UNDER : -

I believe Scott Le BRUN is no longer involved with the VZ and I for one appreciate the service he performed for VZ Users. Well done Scott and we wish you the best for the future.

Harry HUGGINS has stepped in as the new Editor of VZ Down Under and I had the pleasure of meeting him personally while he was in our region on holidays.

VZ USER MARK HARWOOD P.O.BOX 154 DURAL N.S.W. 2158

LE'VZ OOP J.C.E. D'ALTON 39 AGNES St. TOOWONG QLND. 4066
VSOFTWAREZ - SOFTWARE/HARDWARE FOR SALE

VZ DOWN UNDER HARRY HUGGINS 12 THOMAS ST. MITCHAM 3132 VIC.

VZ-LINK - PETER J. HILL P.O.BOX 1972 C.P.O. AUCKLAND N.Z.

WAVZ - GRAEME BYWATER P.O. BOX 388, MORLEY W.A. 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178
SOFTWARE FOR SALE - DISK MENU

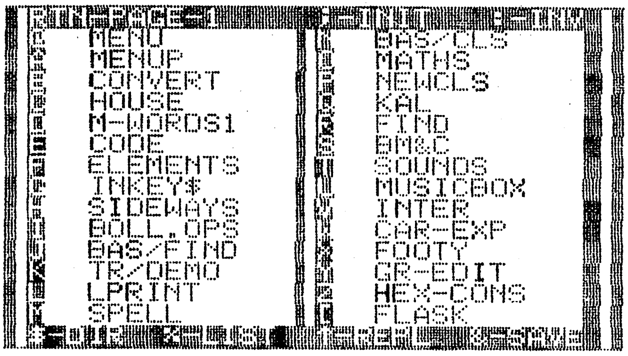
HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND N.S.W. 2299
SECRETARY/EDITOR-JOE LEON (049) 51 2756
PRESIDENT-ROSS WOODS (049) 71 2843

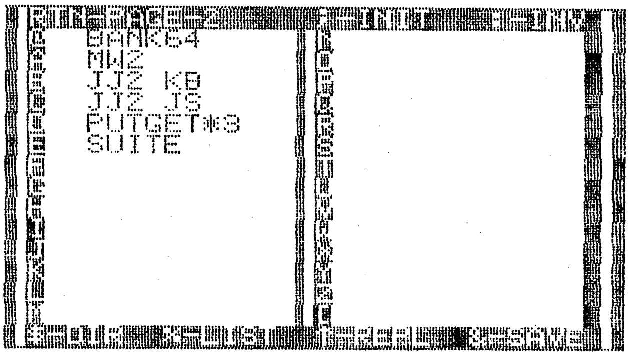
SUBSCRIPTION - H.V.VZ.JOURNAL - 6 MONTHS $9.00 - 12 MONTHS $18.00
New Zealand - 6 MONTHS $12.00 - 12 MONTHS $24.00

MEETINGS - FIRST FRIDAY of MONTH at JESMOND NEIGHBOURHOOD CENTRE
MORDUE PARADE - REAR STOCKLAND MALL (BIG W) JESMOND

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. or NZ 2 Int. Reply Coupons.

Every effort is made to ensure the accuracy of information contained within be it general, technical, programming, etc. No responsibility can be accepted by H.V.VZ.U.G. as a result of applying such information.





The two screen dumps above show two pages of the menu and top and bottom of frame show all relevant commands.

Lines 10 to 15 :-
DO NOT ALTER THEM - Type in exactly as shown. The (:) COLON in front of the command RUN is not a mistake. The program is designed to LOAD and RUN or BRUN basic or machine code programs.

In case of M/C programs the character 'B' is POKED into line 15 replacing the COLON and thereby changing the command RUN into BRUN. The program FILENAME is POKED also into line 15 where the the numbers 1 to 8 are between the quotes. For that reason any attempt to change line 10 and 15 will result in a corrupted program. In case there are less than 8 characters then line 205 puts blank spaces at end of filename to make up 8 characters.

MENU COMMANDS :-

%-LIST - Pressing SHIFT 5 will list lines 500 to 600 where the FILENAME and FILETYPE must be entered into DATA statements seperated by a comma. A 'T' denotes a basic file while a 'B' denotes a M/C file. and need not be typed in as your programs will have different names. Try MENUP,T as your first filename.

RTN-PAGE-1 - The RETURN key toggles between pages 1 and 5 provided each page holds filenames. The current page number is shown.

;-INIT - The (;) key will allow you to INITialise disks and has a QUIT option.

:-INV - This key toggles between NORMAL and INVERSE screens.

$-dir - This key displays the directory and a further function gives you printer printout option.

&-SAVE - This key will save the MENUP program to disk.

^-REPL - If your disk already has a MENUP program on it then SHIFT 'N' will first erase and then save it again using same filename. If you add, change or delete filenames from MENUP then SHIFT 'N' is used to update it.

NOTE - In line 110 "^" denotes SHIFT 'N', the UP ARROW key.

LODING PROGRAMS - With MENUP up and running simply press the required key to the left of the filename and your program will RUN or BRUN depending on type.

```
10 GOTO25
15 :RUN"12345678"
20 '"[illegible]
25 PA=0:DIMP$(140),F$(140)
30 CLS:IFPEEK(30744)<>96THENPOKE30744,96
35 POKE30862,80:POKE30863,52
40 N=N+1:READP$(N),F$(N):IFF$(N)<>"*"THENGOTO40ELSEN=N-1
45 :
50 CLS:COLOR7:POKE30777,1:IFPA*28+1>NTHENPA=0
55 PRINT"[illegible]"CHR$(241+PA)"[illegible]";
60 FOR I=1 TO 14
65 PRINT@(32*I),"[illegible] "CHR$(191+I)"   "P$(I+PA*28);
70 PRINTTAB(15)"[illegible] "CHR$(205+I)"   "P$(I+14+PA*28);TAB(30)"[illegible]";
75 NEXT
80 PRINT@480,"[illegible]";:POKE29183,229
85 POKE30777,35:GOSUB370
95 :
100 X$=INKEY$:X$=INKEY$:IFX$=""THEN100ELSEX=USR(X)
105 IFX$=":",POKE30744,ABS(PEEK(30744)=0)*96:SOUND20,1
110 IFX$="^"THEN250
115 IFX$="&"THEN260
120 IFX$="$"THEN300
125 IFX$=";"THEN400
130 IFX$="%"THENCLS:LIST500-
135 IFX$=CHR$(13)ANDN>28THENPA=PA+1:GOTO50
140 IFX$<"@"ORX$>"["THEN100
145 P=ASC(X$)-63+PA*28:IFP$(P)=""THEN100
150 CLS:PRINT@480,"[illegible] "F$(P)": "P$(P);
155 GOSUB370
195 :
200 IFF$(P)="B"THENPOKE31477,66ELSEPOKE31477,58
205 FORB=1TO8:POKE31479+B,ASC(MID$(P$(P)+"        ",B,1)):NEXT
210 GOTO15
240 :
250 CLS:PRINT@480,"[illegible] T: MENUP";:GOSUB370:ERA"MENUP"
260 CLS:PRINT@480,"[illegible] T: MENUP";:GOSUB370:SAVE"MENUP"
295 :
300 CLS:DIR:STATUS
305 PRINT@17,"[illegible]";:GOSUB370
310 R$=INKEY$:R$=INKEY$:IFINKEY$=""THEN310ELSEX=USR(X)
315 IFR$="Y"THEN340ELSEIFR$="N"THEN50ELSE310
340 IF INP(0)=254 OR INP(0)=246THEN350ELSE355
350 CLS:POKE30876,1:DIR:STATUS:LPRINT:RUN
355 PRINT@480," [illegible]";
360 GOSUB370:SOUND0,9;0,5:RUN
365 :
370 SOUND30,1;25,1:RETURN
375 :
400 PRINT@481," [illegible] ";:GOSUB370
405 I$=INKEY$:I$=INKEY$:IFI$=""THEN405ELSEX=USR(X)
410 IFI$=" "THEN415ELSEIFI$="Q"THEN80ELSE405
415 PRINT@481,"  [illegible]  ";:GOSUB370
420 INIT:GOTO300
495 :
500 DATA MENU,T,MENUP,T,CONVERT,T,HOUSE,T,M-WORDS1,T,CODE,T
505 DATA ELEMENTS,T,INKEY$,T,SIDEWAYS,T,BOLL.OPS,T,BAS/FIND,T
510 DATA TR/DEMO,T,LPRINT,T,SPELL,T,BAS/CLS,T,MATHS,T,NEWCLS,B
520 DATA KAL,T,FIND,B,BM&C,T,SOUNDS,T,MUSICBOX,T,INTER,T
530 DATACAR-EXP,T,FOOTY,T,GR-EDIT,T,HEX-CONS,T,FLASK,T,BANK64,T
560 DATA MW2,T,JJ2 KB,T,JJ2 JS,T,PUTGET*3,T,SUITE,T
600 DATA ,*
```

NOTES FOR PUTGET*3

PUT and GET work from within a BASIC program, using statements based on token 165 for PUT and token 164 for GET. These tokens have to be emplaced in a program, as they are in PUTGET*3 where 'P' and 'G' characters are used to reserve bytes in lines 16 and 18 for the PUT and GET tokens. The number characters '1' and '2' in lines 16 and 18 (following 'P' and 'G') modify the PUT and GET statements to make them PUT1, PUT2, GET1, GET2 statements.

When PUTGET*3 is RUN, you will find that most of lines 16 and 18 are invisible when listed, because your VZ is unable to display the words PUT and GET. The words do not exist for the VZ, only the tokens.

DO NOT TYPE IN the REM lines at the start of PUTGET*3 when you enter the program on your VZ, otherwise RUNing the program will POKE the tokens to the wrong locations and the program won't work.

Line 20 can be rewritten as follows to exclude a RETURN from the subroutine (GOSUB12 or GOSUB14) when the RETURN key is pressed :-

```
20 IFK$=CHR$(13)OR(K$>0ANDK$<7)THEN14ELSERETURN
```

Both PUTGET*3 and INTDATA (see issue 17 of the JOURNAL) set up their routines in the same part of the COMMUNICATIONS REGION, so you cannot have both routines available simultaneously in your VZ.

SUITE is a combined PUTGET*3 and modified INTDATA for disk drive owners, that allows you to use both sets of routines. SUITE sets up and activates the INTERRUPT ROUTINE and four associated subroutines in protected high memory, leaving the COM.REGION location to PUTGET*3. Three of the four subroutines to the INTERRUPT are those of INTDATA :-

<SHIFT> <C> executes CURSOR CLEAR SCREEN.
<SHIFT> <V> executes LDIR:LSTATUS
<SHIFT> <B> executes DIR:STATUS

<SHIFT> <X> now executes a new function (IBS) that allows you to BSAVE screens to disk whenever you want.

IBS saves the whole 2K of video memory. In MODE(1) this will be a single 2K hi res screen. In MODE(0) it will be a single 512 byte text screen and the content of the remainder of video memory. This remainder can be used in MODE(0) to store three text screens using PUT, PUT1 and PUT2 within a BASIC program. Then <SHIFT> <X> will save the current screen and all three stored screens to disk as a single binary file with the file name SCREEN00.

You can use <SHIFT> <X> again and again to save the current content of video memory to disk. Each time, the number at the end of the file name will increment so that your files will save under the file names SCREEN01, SCREEN02, SCREEN03, ..., SCREEN99 then (if you should ever be busy enough with your VZ and disk drive to get this far!) back to SCREEN00.

When you BLOAD a SCREEN file that consists of four MODE(0) screens, you can use GET, GET1 and GET2 to recall the three alternate screens to display.

To use PUT and GET you will need a subroutine in your programs like the one at the start of PUTGET*3: lines 10 to 24 (Issue 19, page 4) .

Immediately after BLOADing a SCREEN file, the INTERRUPT is disabled and will need to be re-enabled before <SHIFT> <X>, <C>, <V>, <B> can be used again. Re-enabling happens automatically in command mode (when a BASIC program is not running) and when a BASIC program is BREAKed, STOPed or ENDed.

The most convenient way to do it without stopping a running program is to execute a SOUND statement from within your program, immediately following the BLOAD statement in the program. If you don't want a noise then use SOUND0,1.

```
5 FORR=1TO217:READA:B=B+A:NEXT
10 IFB<>18830THENPRINT"[illegible]":SOUND30,2:END
14 REM"[illegible]
15 A=(PEEK(30897)+PEEK(30898)*256)-144::B%=A/256:C%=A-B%*256
20 POKE30897,C%:POKE30898,B%:CLEAR50
30 A=PEEK(30897)+PEEK(30898)*256:E=65536:B%=31273
40 FORR=1TO73:READC%:POKEB%,C%:B%=B%+1:NEXT
50 D=A+1-E:FORR=74TO217:READC%:POKED,C%:D=D+1:NEXT
60 D=A+21:B%=D/256:C%=D-B%*256:POKEA+43-E,C%:POKEA+44-E,B%
70 D=A+29:B%=D/256:C%=D-B%*256:POKEA+49-E,C%:POKEA+50-E,B%
80 D=A+1:B%=D/256:C%=D-B%*256:POKE30846,C%:POKE30847,B%
90 POKE30845,195:POKE31107,41:POKE31108,122:POKE31104,72
95 POKE31105,122
99 REM"[illegible]
100 DATA17,0,114
110 DATA183,40,19,254,58,40,15,35,254,49,17,0,116,40,7,254,50
120 DATA17,0,118,32,46,229,33,0,112,24,31,229,33,0,114,17,0,112
130 DATA183,40,21,254,58,40,17,225,35,229,254,49,33,0,116,40
140 DATA7,254,50,33,0,118,32,7,1,0,2,237,176,225,201,225,195
150 DATA151,25
160 REM"[illegible]
170 DATA58,251,104,254,115,40,70,254,121,40,30,254,91,40,93,254
180 DATA122,40,113,201
189 REM"[illegible]
190 DATA34,83,67,82,69,69
200 DATA78,48,48,34,44,55,48,48,48,44,55,55,70,70,0
209 REM"[illegible]
210 DATA33,0,0,205,57,72,33,0,0,126,254,57,48,3,60,119,201,62
220 DATA48,119,43,126,254,57,56,244,62,48,119,201
225 DATA0,0,0,0,0,0
229 REM"[illegible]
230 DATA42,32,120,1,0,114,58,24,120,254,0,62,32,32,2,62,96,119
240 DATA50,60,120,35,121,189,32,236,120,188,32,232,201
249 REM"[illegible]
250 DATA205,196,5,203,71,32,7,62,1,50,156,120,24,15
270 DATA33,230,0,1,75,0,205,92,52,201
279 REM"[illegible]
280 DATA62,0,50,156,120,205,6,73,205,205,82,201
```

ADDITIONS TO DON McKENZIES' PRINTER BUFFER

As a continuation on from the last update on the printer Buffer, in this article I'll describe the latest Eprom and the three new Printed circuit Boards.

The EPROM. The new version Eprom is marked ' PE ' , it has improved diagnostics and a true HEX DUMP MODE and is designed mainly for the three new (main) P. C. Boards. While retaining the Reset, Copy & Test features, there are additional SHIFT, F1 & F2 keys. The Eprom AUTO senses the amount of RAM on board at power-up -: e.g. 64 K or 256 K. All 8 Rams must be installed.

The Boards.
The C3P1 board plugs onto the PBUFF board and is designed for the connection of two more computers. This makes a total of three computers which can feed the PBUFF and are auto polled in a logical sequence (1, 2, 3, 1, 2, 3, 1 ---- etc).
Three DATA LEDS are fitted and indicate at a glance which computer is accessing the printer. The board supports a number of additional features via a set of jumpers which are read by the system at power-up and during run-time.
Some of the features -:

x7 link out - all Data is printed as received by any computer.
x7 link in - a form feed is output between computer changes.
x6 link out - a time delay of approx. 4 seconds takes place on computer input changes.
x6 link in - the delay is set at 8 seconds.

All the links are explained in more detail in the assembly instructions.
The C2 board carries the additional 2 Centronics Sockets for the extra computers and mates up with the C3P1 board via standard 26 way I.D.C. connectors.
The FPIO board takes the place of the original front panel push buttons and toggle switches. Like the C3P1 & PBUFF board, it is Double Sided and this is used to advantage as the components are mounted on one side while the Push buttons and 2 * 7 segment Displays are mounted on the other, the Front as you look at it.
The Display is used to show a number of things -: e.g.
-the current version Eprom while the PBUFF is idle.
-how full the Buffer is as a percentage when loaded & printing.
-the number of copies set or remaining to be printed plus a lot more etc.
The Buttons -:

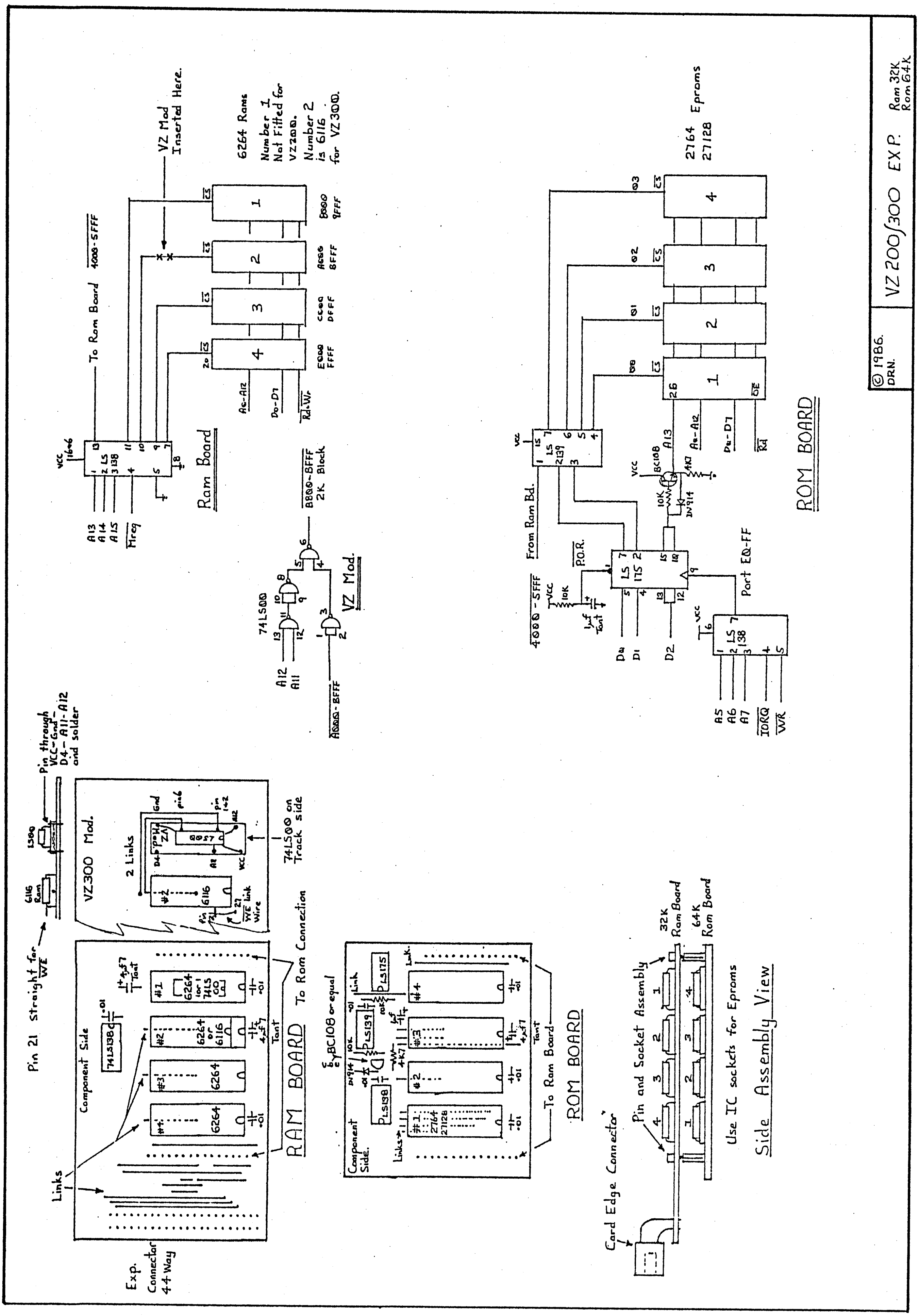
RESET - clears the buffers memory.
TEST - prints the TEST message, runs a diagnostic with any ERRORS sent to the printer.
COPY - prints the Buffers current contents from the last RESET.
SHIFT - used with TEST button.
F1 - with the Buffer idle, it is used with COPY to increment the number of copies in steps of one.
- with the Buffer printing, it is used to PAUSE and stop all output to the printer.
F2 - when used with COPY, it increments the number of copies to be printed in steps of ten.
- when used with RESET, it sets the Buffer into the HEX Mode.

For more and full information / paperwork, write to -:

Mr. DON McKENZIE
29 ELLESMERE CRES.
TULLAMARINE 3043
VICTORIA. AUSTRALIA.

written by
Dave Boyce
JUNE 1988

Addendum to the previous write-up on the Printer Buffer which appeared in the JULY/ AUGUST ISSUE 19 of this magazine . Reading the first paragraph makes the write-up in the MARCH 1987 issue of A.E.M. appear to be written by myself, this is not the case. That article was in fact written by DON McKENZIE (himself), as was an article en-titled ' A MYSTERY SOLVED - how the PBUFF printer buffer project works' which appeared in JANUARY 1988 issue of A.E.M.


VZ 200/300 EXP.
Ram 32K
Rom 64K
© 1986.
DRN.
Ram Board
ROM BOARD
VZ Mod.
6264 Rams
Number 1 Not Fitted for VZ200.
Number 2 is 6116 for VZ300.
VZ Mod Inserted Here.
To Rom Board
4000-5FFF
2764 Eproms
27128
From Ram Bd.
P.O.R.
Port E0-FF
B800-BFFF 2K Block
74LS00
VZ300 Mod.
2 Links
74LS00 on Track side
Pin 21 Straight for WE
Pin through VCC-Gnd-D4-A11-A12 and solder
Component Side
RAM BOARD
To Rom Connection
Links
Exp. Connector 44 Way
BC108 or equal
To Rom Board
Pin and Socket Assembly
Card Edge Connector
Use IC sockets for Eproms
Side Assembly View
32K Ram Board
64K Rom Board

REVIEW by Joe LEON :-

David Newcombe's ram/rom boards have been available for some time although it has'nt been widely known. The ram board is designed to fit inside a memory expansion case and uses it's edge connector which has to be removed from the old board first. The rom board can't be used on it's own as it's been designed to be piggy backed onto the ram board.

In the past David has assembled the boards with memory and or eprom chips as required and you may also be able to purchase just the bare boards like I did. And now to the boards.

32K RAM BOARD :-

It can be used with both VZ 200 and 300. For VZ 200 four 8K rams are used with ram 1 overlaying 4K of internal 6K ram. For VZ 300 only two 8K and a 2K rams are used. In both cases a maximum allowable 34K ram is achieved. T.O.M. is FFFFH (65535), maximum addressable by the Z80.

4K to 64K ROM BOARD :-

Like the ram board, the rom board board can be used with both VZ 200/300 computers. The rom board as is cannot be used with disk drives as it occupies same memory location as the dos rom, 4000H to 5FFFH (8K).

Up to four 4K, 8K or 16K eproms can be used giving from 4K to 64K available to the user. To fit up to 64K in an 8K area bank switching techniques have been used.

The first eprom will have a MENU programmed into it by David. It will also contain a couple of diagnostic programmes for fault finding, some helpful hints, any machine code programmes you might like placed into it, provided of course, that there is sufficient space available. All other eproms will be acessed through the built in menu.

What does it all mean ?

The ram board will expand your VZ 200/300 internal memory to a full 34K of user ram.

The rom board will allow tape based users instant acess to it's built in programs of your choice which you provide for burning into eprom. Some examples are word processors, data bases, etc.

As mentioned before, I purchased just the bare boards for experimentation because they are specifically designed for the VZ. I'm in the process of converting the ram board for 8K sideways ram, 6000H-67FFH area, pseudo 8K Dos and 16K Basic Rom/s which will be battery backed and just by flicking a couple switches revert to a ram pack again.

For prices and availability write to :-

D.NEWCOMBE 928-960 BEENLEIGH - REDLAND BAY ROAD, MSI - 1311
CARBROOK 4130 - QUEENSLAND

PS - Don't forget to include a Stamped Self Addressed Envelope.

FINAL NOTE - David is a very busy man and sometimes his job takes him away from home for months at a time. So if you write to him, please be patient.

```
2 '*************************************************************************
4 '*  60 BY 60 DIGIT MULTIPLY ROUTINE GIVING UP TO 120 DIGITS   *
5 '*  IN ANSWER - 120 BY 120 WITH 240 IN ANSWER ALSO POSSIBLE   *
6 '*  MODIFIED FROM A VIC 20 PROGRAM FOR THE VZ BY N. HUGHES    *
8 '*************************************************************************
10 CLS
20 PRINT"IGNORE PRECEDING ZERO IN ANSWER"
30 FORP=1TO1000:NEXT
80 CLS
90 CLEAR2000:PRINT
110 ' PRINT:INPUT"  ";C$:INPUT"  ";B$:A$="0"+C$
120 INPUT"  ";C$:INPUT"  ";B$:A$=C$
130 CLS
140 GOSUB5000
145 PRINT@8," FINISHED ":PRINT
170 PRINTTAB(2);A$;" * BY ";B$
180 PRINT:PRINT"= ";Y$:PRINT
200 PRINTTAB(2);"IGNORE FIRST ZEROS IN ANSWER":SOUND20,7
215 PRINT:GOTO110
220 END
5000 PF=0:DP=0:LA=LEN(A$)
5005 PRINT@8,"WORKING ON IT"
5010 PF=PF+1:Z$=MID$(A$,PF,1):IFZ$<>"."ANDPF<LATHEN5010
5020 Z$=MID$(A$,PF,1):IFZ$<>"."THENAI$=A$:GOTO5060
5030 I=PF-1:AI$=LEFT$(A$,I)
5040 IFPF=LATHENLA=LA-1:GOTO5060
5050 DP=LA-PF:AI$=AI$+RIGHT$(A$,DP):LA=LA-1
5060 PF=0:LB=LEN(B$)
5070 PF=PF+1:Z$=MID$(B$,PF,1):IFZ$<>"."ANDPF<LBTHEN5070
5080 Z$=MID$(B$,PF,1):IFZ$<>"."THENBI$=B$:GOTO5120
5090 I=PF-1:BI$=LEFT$(B$,I)
5100 IFPF=LBTHENLB=LB-1:GOTO5120
5110 BI$=BI$+RIGHT$(B$,LB-PF):DP=DP+LB-PF:LB=LB-1
5120 GOSUB6000
5145 IFDP=0THEN5170
5150 I=LEN(Y$)-DP
5160 Y$=LEFT$(Y$,I)+"."+RIGHT$(Y$,DP)
5170 RETURN
6000 X$="":Y$=""
6010 FORK=0TOLB-1
6020 CA=0:I=LB-K
6030 FORH=0TOLA-1
6035 J=LA-H
6040 N$=MID$(BI$,I,1):O$=MID$(AI$,J,1):M=VAL(N$)*VAL(O$)+CA:CA=0
6050 IFM>9THENCA=INT(M/10):M=M-CA*10
6060 TC=H+K+1
6070 Q$=STR$(M):IFLEN(X$)<TCTHENX$=X$+RIGHT$(Q$,1):GOTO6120
6080 W$=MID$(X$,TC,1):XI=VAL(W$)+M
6090 IFXI>9THENXC=INT(XI/10):XI=XI-XC*10:CA=CA+XC
6100 LE=TC-1:RI=LEN(X$)-TC
6110 G$=STR$(XI):X$=LEFT$(X$,LE)+RIGHT$(G$,1)+RIGHT$(X$,RI)
6120 NEXTH
6130 S$=STR$(CA):IFCR>0THENX$=X$+RIGHT$(S$,1)
6140 NEXTK
6150 I=LEN(X$)
6160 FORK=0TOI-1
6170 J=I-K:Y$=Y$+MID$(X$,J,1)
6180 NEXTK
6190 RETURN
6200 END
```

```
        BEAM  HEADING
   CENTERED ON NEWCASTLE

LATITUDE    (DEG.MM N OR S)
? 230 S

LONGTITUDE (DEG.MM N OR S)
? 180 N
```

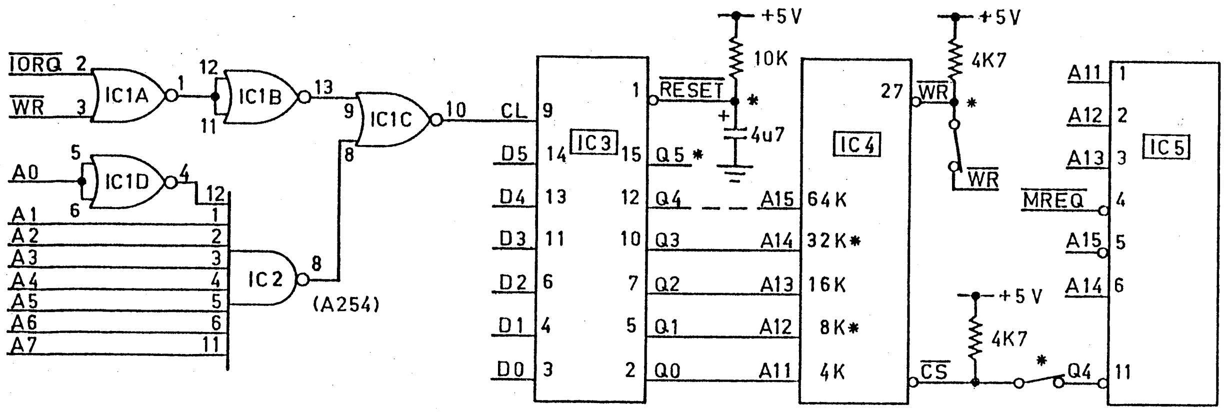
```
   DISTANCE FROM NEWCASTLE
        10577 MILES
        17022 KILOMETRES
BEST BEAM HEADING 138 DEGREES
ANOTHER CALCULATION (Y/N) ?
```

```
10 REM - BEAM HEADING PROGRAMME.
20 CLS
50 A=3.14159/180
60 PRINT@(9+32*2),"BEAM  HEADING"
70 PRINT@(4+32*4)," CENTERED ON NEWCASTLE"
80 T1=-32:T0=-56:T1=T1+T0/60
90 A1=T1*A
100 PRINT
110 T2=151:T3=45:L1=T2+T3/60
120 PRINT
130 INPUT"   LATITUDE   (DEG.MM N OR S)      ";H1$
140 W1=LEN(H1$):W=0
150 FORI=1TOW1:W0$=MID$(H1$,I,1):W=W+1:IFW0$="."THEN160
155 NEXTI
160 B=VAL(LEFT$(H1$,W-1)):B1=(VAL(MID$(H1$,W+1,2))/60)
165 B=B+B1
170 IFRIGHT$(H1$,1)="S"THENB=-B
180 PRINT
190 INPUT"   LONGTITUDE (DEG.MM N OR S)      ";H2$
200 W1=LEN(H2$):W=0
210 FORI=1TOW1:W=W+1:W0$=MID$(H2$,I,1):IFW0$="."THEN220
215 NEXTI
220 L2=VAL(LEFT$(H2$,W-1)):L3=(VAL(MID$(H2$,W+1,1))/60):L2=L2+L3
230 IFRIGHT$(H2$,1)="W"THENL2=-L2
240 GOSUB330
245 CLS
250 PRINT
260 PRINT@(4+32*4),"DISTANCE FROM NEWCASTLE"
265 PRINT
270 PRINTTAB(8)USING"##### MILES  ";D1
275 PRINT
278 PRINTTAB(8)USING"##### KILOMETRES       ";D2
280 PRINT
290 PRINTUSING" BEST BEAM HEADING ### DEGREES ";R2
300 PRINT:INPUT" ANOTHER CALCULATION (Y/N) ";Z0$
310 IFZ0$="Y"THEN10
320 END
```

```
330 REM - CALCULATION ROUTINE.
340 L=L2-L1
350 X=0
360 IFL<-180THEN390
370 IFL>180THEN410
380 GOTO420
390 L=L+360
400 GOTO420
410 L=L-360
420 IFL>=0THENX=1
430 REM - CONV. L & B TO RADIANS.
440 B1=B*A
450 L=L*A
460 REM - DISTANCE ANGLE.
470 P=COS(L)*COS(A1)*COS(B1)+SIN(A1)*SIN(B1)
480 P1=ATN(SQR(1-P*P)/P)
490 P2=P1*180/3.14159
500 REM - ADJ. DIST. ANGLE TO POS
510 IFP2<0THEN530
520 GOTO550
530 P2=P2+180
540 REM - COMPUTE DISTANCE.
550 D1=INT((P2*60*1.15152)+0.5)
560 D2=INT((D1*1.6093)+0.5)
570 REM - COMPUTE BEARING ANGLE.
580 R=COS(B1)*SIN(L)/SIN(P1)
590 IFR>0.99999THENR=0.99999
600 IFR<-0.99999THENR=-0.99999
610 R1=ATN(R/SQR(1-R*R))
620 REM - CONV. TO NEAREST INT.
630 R2=INT((R1*180/3.14159)+0.5)
640 REM - DET QUAD & ADJ DEG.
650 IFABS(R)>0.999998THEN820
660 IFABS(R)<0.00174THEN870
670 B2=(B+0.5)*3.14159/180
680 R3=COS(L)*COS(A1)*COS(B2)+SIN(B2)*SIN(A1)
690 R4=ATN(SQR(1-R3*R3)/R3)
700 R6=COS(B2)*SIN(L)/SIN(R4)
710 IFX=1THEN770
720 IFABS(R6)>ABS(R)THEN750
730 R2=360-ABS(R2)
740 GOTO950
750 R2=180+ABS(R2)
760 GOTO950
770 IFABS(R6)<ABS(R)THEN800
780 R2=180-ABS(R2)
790 GOTO950
800 R2=ABS(R2)
810 GOTO950
820 IFX=1THEN850
830 R2=270
840 GOTO950
850 R2=90
860 GOTO950
870 IFABS(L)>178THEN930
880 IFB<ATHEN910
890 R2=0
900 GOTO950
910 R2=180
920 GOTO950
930 IFB>ATHEN910
940 GOTO890
950 RETURN
960 END
```

4K-64K SIDEWAYS RAM/EPROM FOR 6000H-67FFH RANGE BY JOE LEON - AUGUST 1988

ORIGINAL BY DAVE MITCHELL - 1987



IC1-74LS02 IC2-74LS30 IC3-74LS174 IC4-SEE TEXT IC5-74LS138 & SEE TEXT

The original idea for sideways eprom came from Dave Mitchell and he has had up and running a 4K eprom at 6000H-67FFH, the 2K gap left by the DOS in the cartridge area. It is switched in 2K banks as required. I took the idea to it's logical conclusion by making provision for RAM up to 32K and EPROM up to 64K. That means it's possible to have 32 X 2K banks of memory at your disposal, all banks occupying the same 2K memory location.

In practice I can't see more than 8K of RAM or EPROM being used. Any routines in the 2K gap do slow basic programs down, but it's a small price to pay for added functions available at power up like Dave's Extended Dos which I would'nt be without. And now to the circuit :-

IC-1 and IC-2 together form a decoder for address 254 and the output at pin 10 of IC-1C is used to trigger IC-3.

IC-3 is a latch which is set by the previous stage. The latch is used to select the required bank by switching the relevant RAM/EPROM address lines HI or LO. The Resistor and Capacitor at pin 1 of IC-3 perform a POWER ON RESET functions setting all output latches LO selecting bank (0) at power up.

Pin 1 of IC-3 can be connected to pin 2 of memory expansion connector and in that case the Resistor/Capacitor at pin 1 is no longer required.

IC-5 decodes the range 6000H-67FFH and is the CHIP SELECT for RAM/EPROM.

IC-4 - This needs a little explaining as it can be an 8K or 32K RAM or 4K to 64K EPROM. The pinouts differ on most of them and for that reason are not shown. In case of RAM the Resistor and switch at pin 27 is required and performs WRITE/WRITE PROTECT functions and not required for EPROM.

The CS (CHIP SELECT SWITCH) is used to ENABLE/DISABLE the RAM/EPROM which can be desirable and necessary with the cartridge word processor as it wont work with any RAM/EPROM active at 6000H-67FFH.

Connect all DATA lines to IC-3 and outputs not required can be left unconnected. As only a few IC'S are involved unit can be constructed on VEROBOARD. In my case I mounted the 8K RAM on David Newcombe's ram board and piggy backed the 4 control IC'S using VEROBOARD.

To use is simplicity itself, just load your routines from tape or disk and use the following command to select desired bank :-

OUT 254,N (N=0 to 3 for 8K RAM/EPROM)

```
2 '*****************************************************
4 '* MUSICAL JOYSTICKS BY BRENDON GREGG (10)           *
6 '*****************************************************
10 CLS
20 A=(INP(43) AND 31)
22 IF A=26 THEN SOUND11,1:PRINT"    LEFT-UP    ":GOTO20
24 IF A=22 THEN SOUND 8,1:PRINT"    RIGHT-UP   ":GOTO20
26 IF A=21 THEN SOUND 4,1:PRINT"    RIGHT-DOWN":GOTO20
28 IF A=25 THEN SOUND10,1:PRINT"    LEFT DOWN  ":GOTO20
30 IF A=15 THEN SOUND28,1:PRINT"    FIRE       ":GOTO20
32 IF A= 0 THEN SOUND 5,1:PRINT"    ARM        ":GOTO20
34 IF A=30 THEN SOUND16,1:PRINT"    UP         ":GOTO20
36 IF A=29 THEN SOUND16,1:PRINT"    DOWN       ":GOTO20
38 IF A=27 THEN SOUND23,1:PRINT"    LEFT       ":GOTO20
40 IF A=23 THEN SOUND20,1:PRINT"    RIGHT      ":GOTO20
50 GOTO20
```

FOR PRIVATE SALE . . . .

COMPUTER SHARER SWITCH - Complete ready to plug in, includes paperwork. Requires power supply or plug pack.

PRICE $50.00 & includes P/P within Australia

ORIGINAL D.SMITH SOFTWARE TAPES - $5.00 each plus POST/PACKING

VARIOUS PRINTED CIRCUIT BOARDS - Send for list

PRINTER BUFFER - Needs case plus power supply, else ready to go.

For more info on above send SSAE or phone Dave on (08) 384 6574
Dave BOYCE 41 HEATHER Drive CHRISTIE DOWNS 5164 S.A.

FOR SALE -- E & F W.P. PATCH 3.1

This single Patch will convert your E & F TAPE WORD PROCESSOR for full DISK use while retaining all TAPE functions. It can be used with 1 or 2 DRIVES. Below are the two Menus.

E)DIT TEXT	L)OAD
[illegible]EAR TEXT	S)AVE
[illegible]TEXT	D)IR
	E)RA
	[illegible]EN

STOP PRESS - NEW - STOP PRESS
PATCH 3.3 AVAILABLE SOON. ADDED FUNCTIONS PLUS PRINTER CONTROL CODES IN TEXT. SEE NEXT ISSUE.

Fast SAVI[illegible] [illegible]isk is provided using Block [illegible]

Full instructions are supp[illegible] [illegible]sk transfer utility for your E & F Tape W[illegible]

This Patch will work with V1.0 or V1.2 Disk [illegible] A STATUS facility has been added for V1.0 DOS owners.

FOR MORE INFORMATION CONTACT :-

HUNTER VALLEY VZ USERS' GROUP P.O.BOX 161 JESMOND 2299
N.S.W. AUSTRALIA Phone (049)51 2756

EXTENDED DOS V1.3 - $15.00

This latest version has been updated, improved and with extra commands added.

COMMANDS :-

MERGE - MERGES basic file from disk with program in memory.
DIRA - See example - T:MENU B:PATCH3.1 B:WORDPROC
B:EXTDOS E B:EXTDOS R W:DOS-INST
LDIRA - As above, but to screen and printer.
DIRB - See example - T:MENU 01 00 7AE9 801B 0532
B:PATCH3.1 01 0B 7200 771F 051F
LDIRB - As above, but to screen and printer.
STATUSA - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer, see below.
534 RECORDS FREE 63.500K FREE
OLD - Restores a program after using the NEW command.
OLD. - Prints START, END and LENGHT of program in memory in HEX.
DEC XXXXX - Converts DECIMAL to HEX
HEX XXXX - Converts HEX to DECIMAL
STATUSA and LSTATUSA also works with Version 1.0 DOS.

NEW COMMANDS :-

MENU/FILE COPIER/DISK ORGANISER - $15.00

This utility will read your disk directory and present you with several options. Using the Cursor you can B/RUN any program or select FILE COPY, REN, ERASE, DRIVE 1 OR 2, etc. Besides COPYING TEXT and BINARY files all other files can be copied as well exept DATA files.

DISKFILER - To be released soon

This utility will allow you to keep track of what files are on which disk with Printout option. It's a Database for your disk files.

The above can be purchased from - Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE St. NORTH ROCKHAMPTON QUEENSLAND 4701

For information or demonstration in Newcastle area contact :-
Joe LEON - (049) 51 2756 - 22 DRURY St. WALLSEND NSW 2287

MATTHEW TAYLOR SOFTWARE . . . .

VZ SCREEN ED - $25.00 + $2.00 P&H.

VZ Screen Ed is a new drawing program for HI-RES screens with many powerful options. Both tape and disk users are catered for.

Send your Cheque or Money Order to :-
MATTHEW TAYLOR SOFTWARE -- OR PHONE (049) 75 2350
38 FISHING POINT ROAD RATHMINES NSW 2283 Australia

You can see Matthew at club meetings for a demonstration of Screen ED.

PRODUCED BY H.V.VZ.U.G. . . . 1988

HUNTER VALLEY VZ JOURNAL SUPPLEMENT

RAM COMMUNICATION REGION

BY DAVE MITCHELL MARCH 1987

REVISED BY R.QUINN JULY 1988

30720 - 30740 7800 - 7814 RST JUMP VECTORS

30720-2	7800-2	RST 8 JP 1C96
30723-5	7803-5	RST 10 JP 1D78 (DOS 4293)
30726-8	7806-8	RST 18 JP 1C90
30729-31	7809-B	RST 20 JP 25D9
30732-4	780C-E	RST 28 SET TO RETURN
30735-7	780F-11	RST 30 SET TO RETURN
30738-40	7812-4	RST 38 SET TO EI,RETURN

30741 - 30748 7815 - 781C KEYBOARD D.B.C.

30741	7815	DEVICE TYPE 1
30742-3	7816	DRIVER ADDRESS 12020
30744	7818	INVERSE SCREEN
30745	7819	INVERSE CHARACTERS
30746	781A	? not used
30747-8	781B-C	RAM BUFFER ADDRESS not used

30749 - 30756 781D - 7824 VIDEO D.B.C.

30749	781D	DEVICE TYPE 0
30750-1	781E-F	DRIVER ADDRESS
30752-3	7820-1	CURSOR POSITION range: 28672 to 29183
30754	7822	? not used
30755-6	7823-4	CHECKSUM DURING TAPE LOAD

30757 - 30764 7825 - 782C PRINTER D.B.C.

30757	7825	DEVICE TYPE 6
30758-9	7826-7	DRIVER ADDRESS 1421
30760	7828	LINES PER PAGE ? not used
30761	7829	LINES PRINTED SO FAR ? not used
30762	782A	? not used
30763-4	782B-C	RAM BUFFER ADDRESS ? not used

30765-70	782D-32	? UNUSED BY VZ
30771-3	7833-5	LD A,0 RET PRINTER ERROR VECTOR
30774	7836	START OF KEYBOARD WORK AREA STORE FOR CHARACTER OF KEY PRESSED
30775	7837	?
30776	7838	INVERSE PRINTING 2 enables
30777	7839	LOW RES SPEED UP 35 is slow BIT 3 - VERIFY FLAG (LOADING ROUTINE) BIT 4 - UP/DOWN CURSOR DISABLE BIT 5 - AUTORUN (CRUN) FLAG

30778	783A	?
30779	783B	COPY OF O/P LATCH / BACKGROUND COLOR: 32 green 48 buff
30780	783C	STORES CHARACTER AT CURSOR POSITION
30781	783D	VIDEO CONTROL WORD ? not used
30782-844	783E-7C	USED DURING IPL SEQUENCE -- when computer first switched on

30785	7841	CURSOR FLASH RATE
30786-89	7842-5	USED IN KEYBOARD SCAN ROUTINE
30786	7842	STORES ROW NUMBER OF KEYBOARD: range 1 to 8
30787	7843	STORES COLUMN NUMBER OF KEYBOARD: range 1 to 6
30788-9	7844-5	STORES THE ROW ADDRESS
30790	7846	STORES FOREGROUND COLOR: 0,16,32,48,64,80,96,112 - colors green to orange
30796	784C	LOADING MESSAGE DISABLE FLAG (nonzero = disabled)
30845-7	787D-F	INTERRUPT EXIT
30848-61	7880-D	DIVISION SUPPORT ROUTINE
30862-3	788E-F	USR POINTER
30864-6	7890-2	RND NUMBER SEED
30867-9	7893-5	IN A,0 RET INP ROUTINE
30870-2	7896-8	OUT (0),A RET OUT ROUTINE
30873	7899	LAST KEY PRESSED
30874	789A	ERROR CODE STORAGE/SIGNALS RESUME ENTERED
30875	789B	PRINTER CARRIAGE POSITION
30876	789C	OUTPUT DEVICE CODE 0=video; 1=printer; 255 (-1)=cassette
30877	789D	SIZE OF VIDEO LINE ? not used
30878	789E	HIGH OR LOW RES ? not used
30879	789F	? not used
30880-1	78A0-1	BASIC STACK ADDRESS/ LOWER BOUNDARY OF STRING SPACE
30882-3	78A2-3	CURRENT BASIC LINE NUMBER /ADDRESS OF LINE WITH ERROR
30884-5	78A4-5	START OF PROGRAM
30886	78A6	TAB CURSOR POSITION
30887-8	78A7-8	KEYBOARD BUFFER POINTER: 31208
30889	78A9	0 IF CASSETTE INPUT ELSE NONZERO
30890-3	78AA-D	RANDOM NUMBER SEED
30894	78AE	VARIABLE FLAG
30895	78AF	VARIABLE TYPE FLAG (INTEGER, STRING,SINGLE,DOUBLE)
30896	78B0	EXPRESSION EVALUATION/DATA STATEMENT FLAG
30897-8	78B1-2	TOP OF MEMORY POINTER
30899-900	78B3-4	NEXT LOCATION IN LITERAL STRING POOL
30901-30	78B5-D2	LITERAL STRING POOL: 30 bytes
30931	78D3	LENGTH OF LAST DEFINED STRING
30932-3	78D4-5	ADDRESS OF LAST DEFINED STRING
30934-5	78D6-7	POINTER TO NEXT AVAILABLE PLACE IN STRING SPACE
30936-7	78D8-9	INDEX OF LAST BYTE EXECUTED IN CURRENT STATEMENT EDIT FLAG DURING PRINT USING
30938-9	78DA-B	LINE NUMBER OF LAST DATA STATEMENT READ
30940	78DC	'FOR' STATEMENT FLAG: 1 FOR IN PROGRESS; 0 NO FOR
30941	78DD	INPUT PHASE ENTERED FLAG: 0 during INPUT PHASE
30942	78DE	INPUT/READ FLAG: 0 INPUT; 1 READ
30943-4	78DF-E0	POINTER TO START OF PROGRAM WHEN RUN
30945	78E1	AUTO INPUT FLAG FIELD: 0 = no auto
30946-7	78E2-3	CURRENT LINE NUMBER DURING INPUT PHASE
30948-9	78E4-5	AUTO LINE INCREMENT
30950-1	78E6-7	ADDRESS OF LAST BYTE EXECUTED IN CURRENT LINE
30952-3	78E8-9	BACKSPACED STACK ADDRESS
30954-5	78EA-B	ERROR LINE NUMBE
30956-7	78EC-D	ERROR LINE NUMBER /LINE NUMBER FOR LAST BASIC LINE ENTERED OR DELETED

[illegible]
OR END

30969-70	78F9-A	END OF PROGRAM /START SIMPLE VARIABLES TABLE
30971-2	78FB-C	START OF DIM VARIABLES TABLE
30973-4	78FD-E	START OF FREE MEMORY
30975-6	78FF-900	DATA POINTER (READ STATEMENT)
30977-1002	7901-1A	VARIABLE DECLARATION TABLE 26 entries (A to Z)
31003	791B	TRACE FLAG: 1=on; 0=off
31004	791C	TEMPORARY STORAGE USED BY NUMERIC ROUTINES WHEN UNPACKING A FLOATING POINT NUMBER

31005 - 31012 791D - 7924 WRA1 (WORKING REGISTER AREA)

		INTEGER	SINGLE	DOUBLE	
31005	791D			LSB	
31006	791E			LSB	
31007	791F			LSB	
31008	7920			LSB	
31009	7921	LSB	LSB	LSB	31009-10: USER ARGUMENT STORE
31010	7922	MSB	LSB	LSB	
31011	7923		MSB	MSB	
31012	7924		EXP	EXP	

31013	7925	SIGN OF RESULT DURING MATH & ARITHMETIC OPERATIONS
31014	7926	BIT BUCKET USED DURING DP ADDITION
31015-22	7927-E	WRA2 SAME AS WRA 1
31023	792F	not used
31024-49	7930-49	PRINTER BUFFER
31050-7	794A-51	TEMPORARY STORAGE USED BY DOUBLE PRECISION DIVISION ROUTINE

31058 - 147	7952 - AB	TRS-80 DISK BASIC EXITS — INITIALISED TO RETURNS
31148 - 204	79AC - 79E4	TRS-80 DOS EXIT ADDRESSES
31148	79AC	DOS EXIT FROM 1A1C
	79AF	
	79B2	1AA1
	79B5	1AEC
	79B8	1AF2
	79BB	1B8C
	79BE	2174
	79C1	032C
	79C4	0358
	79C7	1EA6
	79CA	206F
	79CD	
	79D0	2103
	79D3	2108
	79D6	219E
	79DC	222D
	79DF	2278
	79E2	

31205-7	79E5-7	USED BY INPUT
31208-72	79E8-7A28	BASIC LINE INPUT BUFFER
31273 - 31388	7A29 - 7A9C	116 BYTES OF UNUSED MEMORY
31389 - 31404	7A9D - 7AAC	TAPE NAME
31405 - 31446	7AAD - 7AD6	?
31447 - 31462	7AD7 - 7AE6	TAPE LOAD ROUTINE
31463-4	7AE7-8	ZERO

31465	7AE9	START OF USER MEMORY
36864	9000	TOP OF MEMORY VZ-200 (6K)
45184	B800	TOP OF MEMORY VZ-300 (16K)
53348	D000	TOP OF MEMORY VZ-200 (6K+16K)
63488	F800	TOP OF MEMORY VZ-300 (16K+16K)
65535	FFFF	THE VERY TOP